International Journal of Automation and Computing 12(3), June 2015, 337-342
DOI: 10.1007/s11633-014-0870-x

Genetic Algorithm with Variable Length Chromosomes for Network Intrusion Detection

Sunil Nilkanth Pawar[1] Rajankumar Sadashivrao Bichkar[2]

[1]Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra State, India

[2]G. H. Raisoni College of Engineering & Management, Pune, Maharashtra State, India

Abstract: Genetic algorithm (GA) has received significant attention for the design and implementation of intrusion detection systems. In this paper, it is proposed to use variable length chromosomes (VLCs) in a GA-based network intrusion detection system. Fewer chromosomes with relevant features are used for rule generation. An effective fitness function is used to define the fitness of each rule. Each chromosome will have one or more rules in it. As each chromosome is a complete solution to the problem, fewer chromosomes are sufficient for effective intrusion detection. This reduces the computational time. The proposed approach is tested using Defense Advanced Research Project Agency (DARPA) 1998 data. The experimental results show that the proposed approach is efficient in network intrusion detection.

Keywords: Genetic algorithms, intrusion detection, variable length chromosome, network security, evolutionary optimization.

1 Introduction

Any set of actions that attempt to compromise the integrity, confidentiality or availability of resources is called as intrusion[1]. An intruder is an individual or a group of individuals who initiates the actions in the intrusion. An intruder may be a legitimate user of a computer system. It can also be an illegimate user who may enter in an unprotected network service on the computer by exploiting its vulnerability.

An intrusion detection system (IDS) is a monitoring system which reports alarms to the system operator whenever it infers from its detection model. IDS is software, hardware or combination of both used to detect intruder activity. It may have different capabilities depending upon how complex and sophisticated the components are. IDS are manufactured by many companies. An IDS may use signatures, anomaly based techniques or both[2−4]. When IDS detects an intruder, it has to inform security administrator about this using alerts. Alerts may be in the form of pop-up windows, logging to a console, sending e-mail, etc.

Today, use of IDS is considered to be one of the important protection tools. Researchers are working hard to make the IDS smart enough to detect all sorts of attacks. Various soft computing techniques, e.g., fuzzy logic, artificial neural networks and genetic algorithms, are being used for making the intrusion detection rules[5−7].

In a conventional GA, the length of chromosomes is fixed. It makes the GA implementation easy but at the cost of few short comings like:

1) There is no guarantee that all the required rules will be generated.

2) It causes wastage of computational time.

One solution to this problem is to use variable length chromosomes (VLCs)[8] allowing inclusion of one or more rules in chromosomes[9].

This paper presents the use of VLCs in a GA based rule generation for network intrusion detection. This is the first time VLC approach of such a type is being used for intrusion detection problem. These rules are then used for the detection of infected connections. The experimental results show that proposed technique is effective in intrusion detection.

The motivation of the presented work and the brief overview of the IDS are discussed. The remaining paper is organized as follows. Section 2 gives an overview of the genetic algorithm employed in this work. In Section 3, survey of the relevant work is made. Section 4 presents the proposed GA with VLCs. Section 5 presents the implementation and results. Section 6 concludes the paper.

2 Genetic algorithm

Genetic algorithms (GAs) are search algorithms based on the mechanics of natural selection and natural genetics.

GA has been developed by John Holland and his colleagues and students at the University of Michigan. They are different from other optimization techniques in several ways[10]. GA is blind. To perform an effective search for better structures, they only require payoff values. Simplicity of operation and power of effect are the two main attractions of GA approach.

GA has the following three operators[10]: reproduction, crossover and mutation. Genetic algorithm starts with the generation of a random population, then the fitness of the each individual is determined using appropriate fitness until

Regular paper
Manuscript received February 15, 2013; accepted May 5, 2014
Recommended by Associate Editor Matjaz Gams

function. This population undergoes an iterative process solution is found or specified computation is completed. First, the chromosomes are randomly selected using one of the selection techniques such as Roulette wheel selection, tournament selection, rank selection, steady state selection, etc. The selected chromosomes undergo regeneration process. The first step of regeneration is crossover or recombination. There are various crossover techniques such as one-point, two-point, uniform, etc. The result of crossover is the birth of two new chromosomes.

A mutation operator is applied on these newborn chromosomes. Mutation alters one or more gene values in a chromosome. Mutation is an important part of the regeneration process as it helps to prevent the population from stagnating at any local optima. Now the fitness of these chromosomes is determined using the fitness function. When the specified iterations are completed, the best fit chromosome is chosen as the solution for the problem.

3 Related work using GA approach

Different researchers have implemented GA in different ways to generate rules for intrusion detection.

Middlemiss and Dick[11] used GA for weighted feature extraction with specific application to intrusion detection data. They implemented a simple genetic algorithm which evolves weights for the features of data set. A k-nearest neighbor classifier was used for the fitness function of GA as well as to evaluate the performance of the new weighted feature set.

Gong et al.[7] used GA-based approach for network intrusion detection. The genetic algorithm is used to generate the optimized rules for network intrusion detection from network audit data. The support confidence framework is used as fitness function to calculate the fitness of each rule. The fittest rules are then used for network intrusions detection.

Zhao et al.[12] used clustering genetic algorithms to solve the computer network intrusion detection problem. It describes a prototype intelligent intrusion detection system to demonstrate the effectiveness. This system combines two stages into the process including clustering stage and genetic optimization stage. The algorithm can not only cluster the cases automatically, but also detect the unknown intruded action.

Xiao et al.[13] presented a network intrusion detection method based on information theory and genetic algorithm. They used information theory to filter the traffic data and thus reduce the complexity. A linear structure rule is used to classify the network behavior into normal and abnormal behaviors.

Lee et al.[14] presented a feature selection method that maximizes class separation between normal and attack patterns of computer network connections. They have focused on selecting a robust feature subset based on the genetic optimization procedure in order to improve a true positive intrusion detection rate.

Ashfaq et al.[15] used genetic algorithm for generating efficient rules for cost sensitive misuse detection in intrusion detection systems.

Chen et al.[16] designed a training algorithm model based on abnormality detection. The proposed experimental model is based on a hypothesis that if variable x appears more times than the desired value, there is a possibility of occurring abnormality.

In the above papers, the genetic algorithm is used either to generate the detection rules or to select the appropriate features from the data set. They all have used fixed length chromosomes consisting of only one rule in each chromosome. This conventional technique has some drawbacks. First, there is no guarantee that all the required rules will be generated. Further, it causes a lot of wastage of computational time.

4 The proposed GA-VLC based intrusion detection method

The proposed GA-based intrusion detection is implemented in two different phases. In the first phase, the classification rules are generated using a computer algorithm written in Java 6. In the second phase, these rules are used to classify or detect the infected connections.

4.1 Data set

MIT Lincoln Laboratory, under Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) sponsorship, has collected and distributed the first standard data for evaluation of computer network intrusion detection systems. This data is DARPA 1998 data[17]. This data consists of tcpdump and basic security module (BSM) list files. Each line in a list file corresponds to a separate session. Each session corresponds to an individual TCP/IP connection between two computers. The first nine columns in list file provide information which identifies the TCP/IP connection.

Table 1 gives the number of record types that present in the dataset. The first row shows the numbers of normal records. The second and third rows give the distributions of Smurf and Neptune attacks respectively.

The Smurf and Neptune attacks are of Denial of Service type.

Table 1 The distribution of record types

Record type	Number of instances
Normal	45711
Smurf	524
Neptune	15

4.2 Feature selection and representation

Seven most important features having higher possibilities to be involved in network intrusions are selected for defining the intrusion rules[7,18]. These are duration (h: m: s), service (integer), source port (integer), destination port

(integer), source IP (a, b, c and d), destination IP (a, b, c and d), attack name (integer).

Each rule is in if-then form containing a condition and its outcome. The rule is of the form:

IF duration = 0:00:01 & protocol = telnet & source port = 19468 & destination port = 120 & source IP = 001.002.003.004 & destination IP = 172.016.112.050 THEN Neptune. The structure of the chromosome comprising of n rules is shown in Fig. 1.

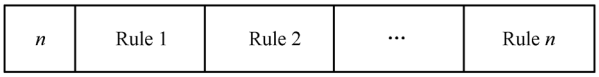

Fig. 1 VLC structure

The number of rules in a chromosome is limited. We begin by defining a particular limit to the number of rules in a chromosome, say 15. But we do not know how many rules are exactly required. This should also be identified by the algorithm. So, the chromosome should be able to increase or decrease the number of rules. We can use wild card values in each field of the rule. We have used wild card values in the third field and fourth part of both source IP and destination IP. We have put −1 in the field chosen for the wild card.

The structure of a rule comprising of genes is shown in Fig. 2. The status field just indicates the presence or absence of an attack.

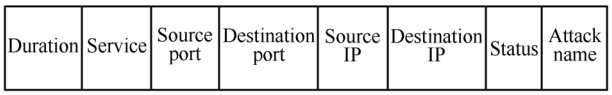

Fig. 2 Rule structure

4.3 Fitness function

The fitness function is based on the amount of errors committed by a rule and the number of rules in a chromosome. Fitness value of a chromosome decreases as the amount of errors committed by its rules increases. Both false positive and false negative errors are considered. False positive error occurs when there is no intrusion occurred but a report of an attack or an attempted attack appears. False negative error occurs when intrusion occurs with no warning. Fitness value also decreases as the number of rules in a chromosome structure increases.

$$f = \frac{k}{1 + error} + \frac{1 - k}{n}$$

where k is chosen to be 0.8, error is the sum of false positive and false negative errors, and n is the number of rules in a chromosome.

4.4 Crossover and mutation

Crossover is an important genetic operator that combines the two parent chromosomes to produce two new offspring chromosomes. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-defined crossover probability.

In the presented approach, one point crossover technique is used. The lengths of both the parent chromosomes are checked and the chromosome whose length is smaller is taken as parent 1. If lengths of both the chromosomes are the same, then any one chromosome is taken as parent 1. Then, a crossover point is randomly chosen for parent 1. As shown in Fig. 3, the part of both the parent chromosomes after the crossover point is interchanged[19, 20].

Mutation occurs on only a few individuals. Each gene in each chromosome is checked for possible mutation by generating a random number between zero and one. If this number is less than or equal to the given mutation probability, i.e., 0.01, then the gene value is changed. Mutations create diversity to search in domain regions that may otherwise be excluded.

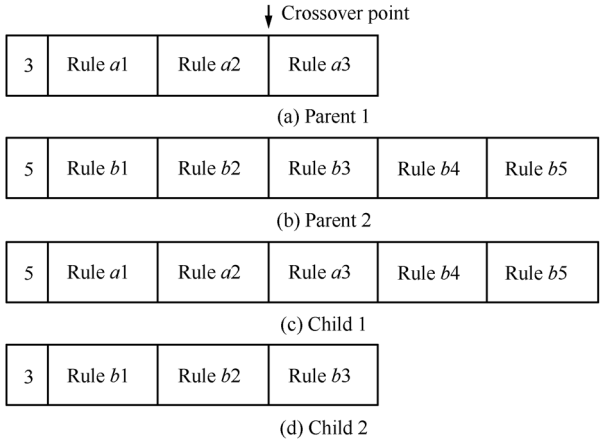

Fig. 3 Crossover on a VLC

5 Implementation and results

The GA with VLCs is implemented using Java language (JDK6). The front end development environment used is NetBeans 7.0. The GA is applied on selected subset of DARPA 1998 data.

The implementation is done in two phases. In the first phase, the classification rules are generated using GA. The number of rules in a chromosome is also determined by GA. Enumeration technique is used to determine the value of each gene for the chromosomes[21]. Normally, while generating the genes, the range of values for each gene is defined and then each gene is generated randomly. We have instead used enumeration technique to determine the value of each gene for the chromosomes. Each gene value from the data set is listed in an ordered fashion. Then, each gene value is randomly chosen out of these listed sets. An effective fitness function is used to calculate the fitness of the chromosomes. After experimentation, the various optimal GA parameters selected were $k = 0.8$, 2000 generations, population of 60, crossover rate of 0.5, one-point crossover and mutation rate of 0.01.

Table 2 Detection rate comparison between the proposed approach and the Gong et al′s. approach[7]

Record type	Proposed GA-VLC approach Detection rate (%)		Gong et al.′s approach[7] Detection rate (%)	
	Training	Testing	Training	Testing
Normal	99.19	98.8	97.2	96.3
Neptune	100	100	96.7	95
Pod	98.79	98.4	96.33	95.2
Smurf	98.45	97.9	95.3	94.1

GA parameters used by Gong et al.[7] were $w_1 = 0.2$, $w_2 = 0.8$, 5000 generations, 500 initial rules in the population, crossover rate of 0.5, two-point crossover and mutation rate of 0.02.

In the presented approach, the maximum number of rules in a chromosome is taken to be 15. The appropriate number of rules is identified by GA.

After generating the classification rules in the first phase, the fittest rule is taken for detection purpose. In the second phase, this rule is used to classify both training as well as testing data set.

We have implemented Gong et al.′s approach[7], and the results obtained are compared with the proposed GA-VLC approach as shown in Table 2.

Implementation is done using a 10-fold cross validation method. In 10-fold cross-validation method, the data set is partitioned into ten parts of equal size, and nine parts of them are used at a time for training and the remaining one is used for testing. The process is repeated ten times, with different partitions used as training data and test data. The most important statistic to collect from each run of algorithm on each data set is the mean of the classification accuracies from ten runs.

Although 10-fold cross validation gives some insight into algorithm performance, the difference is so small that conclusions cannot be made objectively. Hence, a statistical test is conducted. As the input data is normally distributed, small sample paired t-test using MINITAB software is conducted. In this test, measures of algorithm performance on every fold are taken as an input. We observe that P-value (0.000) is less than the alpha (α) level (5%). We reject null hypothesis as the difference is greater than zero (positive), i.e., there is significant difference in the detection rate of the proposed GA-VLC approach and the Gong et al.′s approach[7].

As the GA runs progresses, the accuracy of intrusion detection generally improves until maximum accuracy is obtained. Often, GA may also land in local maxima unless GA parameters are properly set. Table 3 shows the percentage detection for different number of GA generations for the population size of 500 using Gong et al.′s approach.

Table 4 shows the percentage detection for different numbers of GA generations for the population size of 60 using GA-VLC approach.

As shown in Fig. 4, as the number of generations is increased, the detection rate is improved. In Gong et al.′s approach[7], good results are obtained after 5000 generations In GA-VLC approach, the best results are achieved only after 2000 generations.

Table 3 Number of generations against detection accuracy (Using Gong et al.′s approach[7])

Generations	Detection accuracy (%)
500	52.0
1000	58.7
1500	61.6
2000	69.0
2500	72.2
3000	78.0
3500	82.0
4000	85.5
4500	90.3
5000	95.1

Table 4 Number of generations against detection accuracy (Using GA-VLC approach)

Generations	Detection accuracy (%)
500	62.0
1000	78.7
1500	86.6
2000	98.7

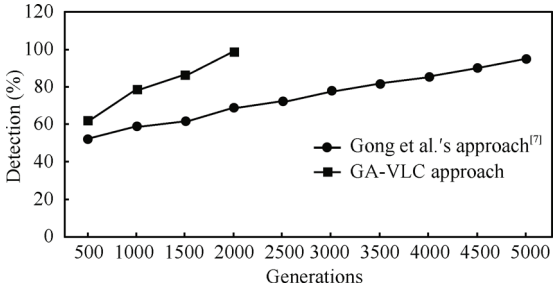

Fig. 4 Effect of generations on detection accuracy

Further, GA-VLC results are compared with various algorithm results used for building decision trees, such as GATree, J48 and CART. For GATree implementation, GATree software[22] is used. J48 and simple CART algorithms are implemented in the open source software called Weka[23]. Implementations are done on 10% KDD Cup 1999 data[24]. For all implementations, 10-fold cross validation technique is used. Results obtained with decision tree algo-

rithm are compared with the proposed GA-VLC algorithm results as shown in Table 5.

Table 5 Comparison of proposed GA-VLC algorithm results with decision tree algorithm results

Classification accuracy (%)			
GA tree	J48	CART	Proposed GA-VLC approach
98.72	99.96	99.87	98.78

Table 6 compares the detection rate of proposed GA-VLC approach with the detection rate of other approaches.

Table 6 Detection rate comparison of proposed approach with other approaches

Approach	Detection rate (%)
Weiming Hu et al.[25]	91.15
Nannan Lu et al.[26]	97.54
Chi Cheng et al.[27]	98.81
H. Altwaijry[28]	99.36
Y. Li et al.[29]	98.62
Proposed approach	98.78

Hu et al.[25] proposed online Adaboost-based intrusion detection algorithms, in which decision stumps and online Gaussian mixture models (GMMs) were used as weak classifiers for the traditional online Adaboost and the proposed online Adaboost. They give 90.13% and 91.15% detection rate. Lu et al.[26] proposed an integrated fuzzy GNP rule mining with distance based classification which yielded 97.54 % detection rate. Cheng et al.[27] proposed a basic extreme learning machine (ELM) method based on random features and a kernel based ELM method for classification. By using kernel based ELM, a good detection rate of 98.81% is achieved. Altwaijry[28] developed an intrusion detection system based on Bayesian probability. The Bayesian classifier was able to detect intrusion with a detection rate of 99.36%. Li et al.[29] proposed an efficient intrusion detection system based on support vector machines and gradually feature removal method. It achieves 98.62% detection accuracy.

6 Conclusions

In this paper, an effective GA-based technique is presented for intrusion detection. It has used VLCs. An enumeration technique is used in genetic algorithm framework for the generation of classification rules. This reduces the search space and provides a good speed-up.

In the presented approach, maximum number of rules in a chromosome is taken to be 15. The appropriate number of rules is identified by GA. In the Gong et al.′s approach, the top 20 best quality rules were taken as the final classification rules. So, it is evident that the number of rules used in the presented approach is less. This reduces the computational time.

Results presented in Table 2 prove that percentage detection rate obtained by the proposed GA-VLC approach is better than the Gong et al.′s approach[7].

As the number of generations is increased, the detection rate is improved. In Gong et al.′s approach[7], the best results are obtained after 5000 generations, where as in GA-VLC approach, the best results are achieved only after 2000 generations. As the computational time is directly proportional to the number of generations, a substantial time is saved using GA-VLC approach.

As presented in Table 5, classification accuracy obtained with J48 and CART decision tree is extremely good. The classification accuracy obtained with proposed algorithm is a bit better than GATree algorithm.

From experimental results, it is evident that the proposed technique is effective in network intrusion detection. Because it provides better result than Gong et al.′s approach[7] even while using smaller number of classification rules.

From Table 6, it is evident that the results obtained by using the proposed GA-VLC approach are comparable with other approach results.

References

[1] S. Mukkamala, J. Guadalupe, A. Sung. Intrusion detection using neural networks and support vector machines. In *Proceedings of the International Joint Conference on Neural Networks*, IEEE, Honolulu, HI, USA, vol. 2, pp. 1702–1707, 2002.

[2] S. Owais, V. Snasel, P. Kromer, A. Abraham. Survey: Using genetic algorithm approach in intrusion detection systems techniques. In *Proceedings of the 7th Computer Information Systems and Industrial Management Applications*, IEEE, Ostrava, USA, pp. 300–307, 2008.

[3] D. J. Day, Z. X. Zhao. Protecting against address space layout randomisation (ASLR) compromises and return-to-libc attacks using network intrusion detection systems. *International Journal of Automation and Computing*, vol. 8, no. 4, pp. 472–483, 2011.

[4] M. Arun, A. Krishnan. Functional verification of signature detection architectures for high speed network applications. *International Journal of Automation and Computing*, vol. 9, no. 4, pp. 395−402, 2012.

[5] J. Gomez, D. Dasgupta. Evolving fuzzy classifiers for intrusion detection. In *Proceedings of 2002 IEEE Workshop on Information Assurance*, IEEE, West Point, NY,USA, pp. 321–323, 2002.

[6] M. Moradi, M. Zulkernine. A neural network based system for intrusion detection and classification of attacks. In *Proceedings of IEEE International Conference on Advances in Intelligent Systems-theory and Applications*, IEEE, Luxembourg, Amsterdam, pp. 148–153, 2004.

[7] R. H. Gong, M. Zulkernine, P. Abolmaesumi. A software implementation of a genetic algorithm based approach to network intrusion detection. In *Proceedings of the 6th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and the 1st ACIS International Workshop on Self-Assembling Wireless Network*, IEEE, Washington, DC, USA, pp. 246–253, 2005.

[8] J. H. B. Ang, K. C. Tan, A. A. Mamun. A memetic evolutionary search algorithm with variable length chromosome for rule extraction. In *Proceedings of IEEE International*

Conference on Systems, Man and Cybernetics, IEEE, Singapore, pp. 535–540, 2008.

[9] R. Cattral, F. Oppacher, D. Deugo. Rule acquisition with a genetic algorithm. In *Proceedings of the Congress on Evolutionary Computation*, IEEE, Washington, DC, USA pp. 125–129, 1999.

[10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, 7th ed., Hong Kong, China: Pearson Education, pp. 1–23, 2004.

[11] M. J. Middlemiss, G. Dick. Weighted feature extraction using a genetic algorithm for intrusion detection. In *Proceedings of Congress on Evolutionary Computation*, IEEE, Canberra, ACT, Australia, pp. 1669–1675, 2003.

[12] J. L. Zhao, J. F. Zhao, J. J. Li. Intrusion detection based on clustering genetic algorithm. In *Proceedings of International Conference Based on Machine Learning and Cybernetics*, IEEE, Guangzhou, China, pp. 3911–3914, 2005.

[13] T. Xiao, G. Z. Qu, S. Hariri, M. Yousif. An efficient network intrusion detection method based on information theory and genetic algorithm. In *Proceedings of the 24th IEEE International Performance Computing and Communications Conference*, IEEE, Phoenix, AZ, USA, pp. 11–17, 2005.

[14] C. H. Lee, S. W. Shin, J. W. Chung. Network intrusion detection through genetic feature selection. In *Proceedings of 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, IEEE, Las Vegas, NV, USA, pp. 109–114, 2006.

[15] S. Ashfaq, M. U. Farooq, A. Karim. Efficient rule generation for cost-sensitive misuse detection using genetic algorithms. In *Proceedings of International Conference on Computational Intelligence and Security*, IEEE, Guangzhou, China, vol. 1, pp. 282–285, 2006.

[16] Z. M. Chen, J. Y. Feng, S. Xu, R. Z. Xu. The research of intrusion detection technology based on genetic algorithms. In *Proceedings of International Conference on Networks Security, Wireless Communications and Trusted Computing*, IEEE, Wuhan, China, pp. 248–250, 2009.

[17] MIT Lincoln Laboratory. *DARPA datasets*, MIT, USA, [Online], Available: http://www.ll.mit.edu/IST/ideval/data/data_index.html, Dec., 15, 2012.

[18] W. Li. A Genetic Algorithm Approach to Network Intrusion Detection, SANS Institute, USA, 2004.

[19] R. Rajesh, M. R. Kaimal. GAVLC: GA with variable length chromosome for the simultaneous design and stability analysis of T-S fuzzy controllers. In *Proceedings of IEEE International Conference on Fuzzy Systems*, IEEE, Hong Kong, China, pp. 1389–1396, 2008.

[20] B. Hutt, K. Warwick. Synapsing variable-length crossover: Meaningful crossover for variable-length genomes. *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 118–131, 2007.

[21] S. N. Pawar, R. S. Bichkar. Using enumeration in a GA-based intrusion detection. *International Journal of Computer Applications*, vol. 56, no. 15, pp. 44–48, 2012.

[22] A. Papagelis, D. Kalles. GA Tree: Genetically evolved decision trees. In *Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, Vancouver, BC, Canada, pp. 203–206, 2000.

[23] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten. The WEKA data mining software: An update. In *Proceedings of ACM SIGKDD Explorations Newsletter*, ACM, New York, USA, vol. 11, no. 1, pp. 10–18, 2009.

[24] ACM KDD CUP 1999. *Datasets*, [Online], Available: http://kdd.ics.uci.edu/Databases/kddcup99/kddcup, December 15, 2012.

[25] W. M. Hu, J. Gao, Y. G. Wang, O. Wu, S. Maybank. Online adaboost-based parameterized methods for dynamic distributed network intrusion detection. *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 66–82, 2014.

[26] N. N. Lu, S. G. Mabu, T. Wang, K. Hirasawa. Integrated fuzzy GNP rule mining with distance-based classification for intrusion detection system. In *Proceedings of International Conference on Systems, Man, and Cybernetics*, IEEE, Seoul, Korea, pp. 1569–1574, 2012.

[27] C. Cheng, W. P. Tay, G. B. Huang. Extreme learning machines for intrusion detection. In *Proceedings of the International Joint Conference on Neural Networks*, IEEE, Brisbane, QLD, Australia, pp. 1–8, 2012.

[28] H. Altwaijry. Bayesian based intrusion detection system. *IAENG Transactions on Engineering Technologies*, Netherlands: Springer, pp. 29–44, 2013.

[29] Y. H. Li, J. B. Xia, S. L. Zhang, J. K. Yan, X. C. Ai, K. B. Dai. An efficient intrusion detection system based on support vector machines and gradually features removal method. *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

Sunil Nilkanth Pawar graduated from Marathwada University, India in 1993. He received the M. Tech degree from Centre for Electronics Design and Technology of India (CEDTI), India in 2000. He is currently an associate professor in Electronics & Telecommunication Engineering Department in Jawaharlal Nehru Engineering College, India.

His research interests include genetic algorithms and network security.

E-mail: nil_pawar@yahoo.com (Corresponding author)

ORCID iD: 0000-0002-7044-6329

Rajankumar Sadashivrao Bichkar obtained the B. Eng. and M. Eng. degrees in electronics from the Shri Guru Gobind Singhji Institute of Engineering and Technology, India in 1986 and 1990, respectively, received the Ph. D. degree from India Institute of Technology Kharagpur in 2000. He served as a faculty member in the Computer Engineering and Electronics Engineering Departments in the SGGS Institute of Engineering and Technology from 1986 to 2007. Now, he is a professor in the Department of Electronics and Telecommunication Engineering, G. H. Raisoni College of Engineering and Management, India.

His research interests include application of genetic algorithms to various search and optimization problems in electronics and computer science.

E-mail: bichkar@yahoo.com

ORCID iD: 0000-0002-6741-9178